AMENDMENTS TO THE CLAIMS

1. – 8. (Canceled)

9. (Original) A method of accessing data, comprising the steps of:

receiving first data;

incrementing a first register containing a count value in response to said first data to provide an incremented count value; and

storing, in response to a first condition of a flag, (i) said incremented count value in a second register and (ii) said first data in a memory.

10. (Original) The method according to claim 9, further comprising a step of setting said flag to a second condition in response to said first data.

11. (Original) The method according to claim 9, further comprising the steps of:

reading a first data value stored in said second register;

reading said first data from said memory;

reading a second data value stored in said second register and comparing said first and second data values.

12. (Original) The method according to claim 11 further comprising a step of processing said first data in response to a result of said comparing step.

13. (Currently Amended) The method according to claim 11, further comprising the step of resetting a condition of said flag only if said second data value matches the first data value stored in said second register.

14. (Original) The method according to claim 9, further comprising a plurality of steps of reading portions of said first data from said memory and steps of comparing values read from said second register and, in response, selectively processing said first data.

15. (Original) The method according to claim 9, further comprising the steps of:

setting said flag to a second condition in response to said first data;

reading, on plural occasions, different portions of said first data from said memory;

comparing values stored in said second register prior to and after said reading step and, in response, selectively processing said first data stored in said memory; and

in response to said comparing step resetting said flag back to said first condition only if the value read from said second register for use in comparing step matches the value stored in said second register.

16. (Original) The method according to claim 15, further comprising the steps of:

detecting a non-equivalence of said values and, in response, inhibiting a processing of said first data stored in said memory.

17. (Original) The method according to claim 9 wherein said first data includes error information and said count value includes a number of error events detected.

18. – 20. (Canceled)

21. (Previously Presented) A method, comprising the steps of:

providing a token which can be atomically read and which uniquely identifies an error log entry which cannot be atomically read and evaluated for change; and

clearing said error log entry using said token as a key.

22. (Previously Presented) The method according to claim 21 further comprising the steps of:

storing error data as said error log entry and updating said token to correspond to said error data.

23. (Previously Presented) The method according to claim 22 further comprising the steps of:

reading said error data using said token to validate said error data.

24. (Previously Presented) The method according to claim 21 wherein said token includes an indication of an ordinality of said error log entry.

25. (Previously Presented) The method according to claim 21 wherein said token includes an indication of a status of said error log entry.

26. (Previously Presented) The method according to claim 21 further comprising ensuring only valid copies of error data are obtained corresponding to said error log entry and inhibiting clearing of unrecorded data corresponding to said error log entry.

27. (Previously Presented) The method according to claim 21 further comprising a step of forming a digital signature of said error log entry to create said token.

28. (Previously Presented) The method according to claim 21 further comprising a step of hashing said error log entry to create said token.

29. (Currently Amended) A system for accessing data comprising:

a count register which maintains a count of occurrences of an error in the system;

a status register which maintains a status of the system, wherein the status is one of [[on]] clear and error; and

a tag register which stores the value of the count register if the status register is clear.